Context as the building blocks of meaning

A retrieval model for the semantic representation of words

Peter J. Kwantes

Defence R&D Canada–Toronto

Technical Report

DRDC Toronto TR 2003-061

April 2003

Author

Peter J. Kwantes

Approved by

Dr Lochlan Magee

Head, Simulation & Modelling for Acquisition, Rehearsal and Training (SMART) Section

Approved for release by

K. M. Sutton

Chair, Document Review and Library Committee

Abstract

A computational model of semantic memory is described. Based on simple principles borrowed from a computational account of episodic memory, it is shown that a memory model that is exposed to a large corpus of language can develop representation for words that look like their ‘meanings’.

Résumé

Ce rapport décrit un modèle informatique de mémoire sémantique. Partant de simple principes empruntés à la description computationnelle d’une mémoire épisodique, on démontre qu’un modèle de mémoire exposé à un vaste corpus de mots peut former une représentation de mots qui ressemble au sens de ces mots.

This page intentionally left blank.

Executive summary

In this report, a computational model of human semantic memory is described. The model borrows heavily from a popular account of episodic memory by Hintzman (1984) in which experiences are represented in memory as individual traces. A simulation of the model shows that despite not knowing anything about the meanings of words, the model is capable of constructing a vector representation of meaning. The model is evaluated on its ability to create meanings by comparing the meaning vectors of a set of words (representing four semantic categories) to each other. Using multidimensional scaling, it is shown that the vector representations for words within a category are more similar to each other than vectors representing words between categories. The model is then discussed in terms of its usefulness as a potential tool for categorizing machine-readable documents.

Sommaire

Dans ce rapport, on décrit un modèle de mémoire sémantique humaine. Le modèle puise considérablement dans un compte rendu de mémoire épisodique par Hintzman (1984) où les expériences sont représentées dans la mémoire comme des empreintes distinctes. Une simulation du modèle montre que malgré sa complète ignorance du sens des mots, le modèle peut créer une représentation vectorielle du sens. On l'évalue en fonction de sa capacité de créer un sens en comparant les vecteurs de sens d'une série de mots (représentant quatre catégories sémantiques) entre eux. Une analyse multidimensionnelle démontre que la ressemblance est plus étroite entre les représentations vectorielles de mots de la même catégorie qu'entre les représentations vectorielles de mots de catégories différentes. La discussion concernant le modèle est exposée en fonction de l'utilité de ce dernier en tant qu'outil potentiel servant à classer des documents lisibles par machine.

Peter J. Kwantes. 2003. Context as the building blocks of meaning: A retrieval model for the semantic representation of words. DRDC Toronto TR 2003-061. DRDC Toronto.

This page intentionally left blank.

Table of contents

List of figures

Acknowledgements

Some of the work contained in this paper was supported by an Early-Career research grant awarded to the author from the University of Queensland. I am indebted to Mike Humphreys and Simon Dennis for useful discussions during the course of this work. Correspondence concerning this report should be addressed to Peter Kwantes at Defence R&D Canada – Toronto, PO Box 2000, 1133 Sheppard Ave. W., Toronto, Ontario, Canada, M3M 3B9. Electronic mail can be sent to peter.kwantes@drdc-rddc.gc.ca

This page intentionally left blank.

Introduction

How do people learn the meanings of the words they encounter during their exposure to language? More pointedly, how is it that we seem to know so much about language despite what seems to be a relatively limited exposure to it? Landauer and Dumais (1997) offered one answer to the question in a computational model called Latent Semantic Analysis (LSA).

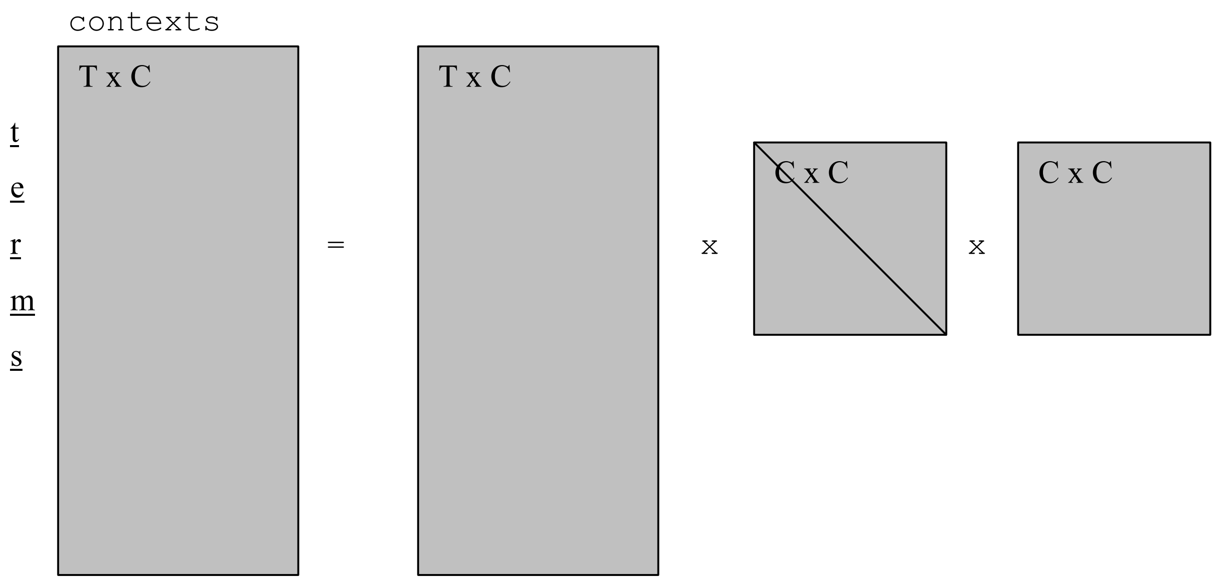
A word's meaning in LSA is a vector describing the frequency with which the word occurs in potentially thousands of contexts or documents. The basic idea behind the model is that words that have similar meanings will tend to appear in the same or similar contexts (see Burgess & Lund, 1995 for a model that works on similar principles). For example, key words appearing in documents about automobiles tend not to appear in documents about telephones.

How does LSA work? The system starts by "reading" thousands of documents. For each document (which could be an encyclopaedia entry or newspaper article), it tabulates the number of times each word in the document appears. To maintain word frequency information over several thousand documents, a term-by-document matrix is formed in which each word is described as a vector, the elements of which contain the number of times the word occurred in each document. One can view each word's vector as a description of a word as existing in N-dimensional space, where N is the number of documents, or contexts, in the training corpus.

If words with similar meanings always occurred together in the same documents, the vector describing the documents in which a word appears would serve as an adequate and reliable characterization of its meaning. Such "local" co-occurrences, however, fail to capture a deeper sense in which two words can be related. For example, the terms, <u>Toronto</u> and <u>Brisbane</u> have a clear semantic similarity by virtue of the fact that they are both cities, they are both capitals, have a parliament, are run by a premier, *et cetera*. Despite any similarity the terms may have in semantic space, there is no reason why Toronto and Brisbane should ever occur in the same document. If the two never occur together in the same document, a term-by-document matrix will not pick up their similarity. What is needed, is a mechanism that can exploit higher-order associations between the two terms. For example, the system needs to know that Toronto and Brisbane are related because, even though the terms never occur in the same document, they appear in documents that share terms associating them like <u>city</u>, <u>capital</u>, and <u>parliament</u>. In the paragraphs to follow, I describe, in fairly general terms, how LSA extracts semantic relationships from such higher-order similarities between words.

LSA applies a statistical technique that is similar to Principle Components Analysis called Singular Value Decomposition (SVD) to the term-by-document matrix. SVD decomposes the term-by-document matrix into three matrices. As is shown in Figure 1, a $t \times c$ sized matrix **A** where $c < t$ can be decomposed into (a) an $t \times c$ column-orthogonal matrix **B**, (b) the transpose of an $c \times c$ matrix **C** of row orthogonal values, and (c) an $c \times c$ diagonal matrix **D** of singular values composed of non-zero values.

If the three matrices are multiplied together, the original matrix (save for a bit of rounding error) is perfectly reconstructed. The real key to the reconstruction of the original matrix lies in the diagonal matrix of the decomposition. The diagonal matrix **D** represents how the **B** and **C** matrices of row- or column-orthogonal vectors are related to each other to re-create the



Then, after dimension reduction to *m* dimensions:

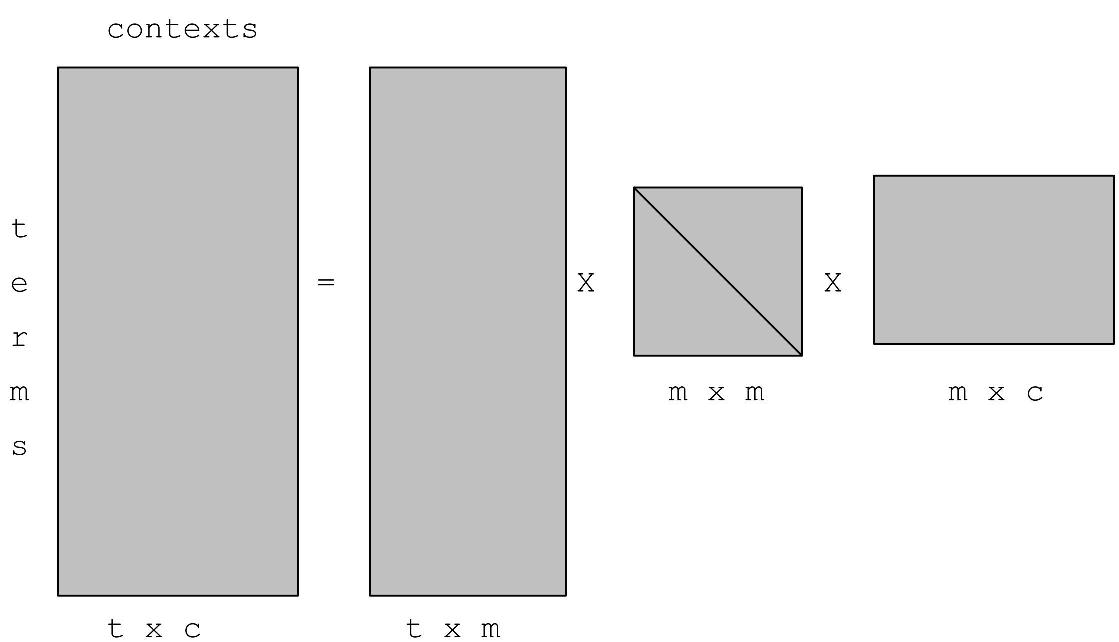


Figure 1. An Illustration of Singular Value Decomposition

original term-by-document matrix. More specifically, each value of **D** represents one orthogonal dimension on which the values of the original t X c matrix differ. The magnitude of each value is an indication of the dimension's salience. That is, the larger the magnitude of a value on the diagonal, the more important that dimension is in recreating the original matrix.

To transform a word's vector into one resembling a meaning, an arbitrary number of the most salient dimensions in the diagonal matrix are maintained, while the remaining ones are set to zero (in the simulations reported by Landauer and Dumais (1997), between the top 200 and 300 dimensions were maintained). Then, the three component matrices of the SVD are multiplied together to recreate the original term-by-document matrix. By selecting the top few hundred dimensions however, the dimensionality of the component matrices are changed; now, instead of the terms differing on c dimensions, they are forced to differ on m (see the bottom part of the illustration in Figure 1)

Because **D** no longer contains <u>all</u> the dimension information needed to recreate an exact copy of the original matrix, the reconstituted matrix is an approximation to the original based on the remaining dimensions. LSA uses the information it has in the remaining dimensions to call upon higher-order relationships between words to fill in the cells of the word's vector. Because higher-order relationships are exploited, Toronto becomes related to Brisbane because, despite the fact the two terms may never have appeared in the same document, the documents they appear in share other words like city, capital, and so on.

How well does the technique work? LSA's power has been demonstrated in a variety of domains. Thus far, it has been able to perform as well as a foreign student on the Test of English as a First Language (TOEFL; Landauer & Dumais, 1997), classify documents in a meaning-based query system (Dumais, 1994), match reviewers to submitted papers (Dumais & Nielson, 1992), and simulate some semantic priming data collected in the laboratory (Landauer, Foltz, & Laham, 1999),

Landauer and Dumais (1997) were quick to point out that they did not believe that the brain performed SVD on co-occurrence information stored in memory. They did claim however, that whatever psychological mechanisms are involved in creating semantic representations, it does something similar to what is accomplished by SVD. In this paper, I introduce a model of semantic representation that borrows some ideas from a well-known computational account of human performance in episodic memory tasks. As such, the model is introduced as a first stab at a psychologically plausible mechanism that accomplishes much the same thing as SVD.

The model

The model's architecture borrows some ideas from Minerva2 (Hintzman, 1984; 1986; 1988). However, whereas Minerva2 was designed to explain memory phenomena in episodic memory tasks, the model I describe next extends the basic ideas behind the model to semantic memory.

Imagine that for every word you encountered, a trace (represented as a vector of features) of it was laid down in memory. In addition to the features that describe the word (which are not described in this article), each vector also contained features uniquely describing the context in which you learned/encountered it. Suppose further that each time you encountered the same word, the vector describing it and its context was summed to the existing one.

Figure 2 illustrates what the memory system looks like after committing three small documents to memory (function words have been excluded from the example). Each trace holds information about the contexts in which the word appeared. I refer to the vector describing the contexts in which a word occurred as the *context vector*. Words that appear in the same context will share have the same context vector. Likewise, the same word occurring in different contexts have different context vectors.

WORD	DOC 1	DOC 2	DOC 3
BIG	2	0	0
TRUCKS	1	0	0
WITH	1	0	0
TIRES	1	0	0
TORONTO	0	1	0
CAPITAL	0	1	1
ONTARIO	0	1	0
BRISBANE	0	0	1
STATE	0	0	1

Document 1: Big trucks with big tires.

Document 2: Toronto is the capital of Ontario

Document 3: Brisbane is a state capital

Figure 2. The contents of memory after learning three documents

How is meaning information retrieved from the model? I treat the retrieval of meaning as a two-stage process. In the first stage, identified characters of a word are used as a probe to retrieve a word's identity (its spelling and phonology) from memory. In addition to the word's identity, the context falls out of memory as well. For example, in the three-document example in Figure 2, the word capital is represented by the vector, [0 1 1] because it only occurred in the second and third document/context. Hence, the composite context vector is a description of all the contexts in which the word occurs.

After the composite context vector has been retrieved, the system has the information it needs to retrieve the meaning of the word. In essence, the model asks the question, "What other words in my memory share contexts with the word I just retrieved"?

Meaning retrieval involves using the composite context vector as a retrieval probe to retrieve a copy of itself from memory.

The composite context vector is applied to, and resonates with every context vector in memory. The extent to which a context vector in memory resonates with, or is activated by, the probe is a function of their similarity. I measure similarity as the vector cosine between the two:

$$S = \left[\frac{\sum_{i=1}^{N} P_i \times T_i}{\sqrt{\sum_{i=1}^{N} P_i^2} \times \sqrt{\sum_{i=1}^{N} T_i^2}} \right]^{\alpha}$$

Where P and T correspond to the context information contained in the probe and memory traces, respectively, and N represents the total number of contexts contained in memory. The α value can be adjusted to control the activation of context vectors that are an imperfect match to the probe. As α increases, the activation of imperfect matches in response to the probe vector decreases (in the simulation to follow, the parameter was set to 1).

After the context vectors in memory are activated, elements of each trace vector are multiplied by their activations (*A*) according to the formula:

$$T_{i,j} = T_{i,j} \times S$$

After their activation, the elements of the activated trace vectors are summed across all the traces in memory to form a composite of the probe vector. The formula for creating the composite vector, C, is:

$$C_j = \sum_{i=1}^{Ntraces} T_{i,j}$$

The composite vector that is retrieved from memory can be thought of as a representation of the meaning of the word. For example, consider the terms <u>Brisbane</u> and <u>Toronto</u> in the small memory system described in Figure 2. Notice that the two terms never occur in the same document. Hence, at the level of the term-by- document matrix formed during encoding, <u>Toronto</u> and <u>Brisbane</u> are orthogonal concepts. When the context vector for <u>Brisbane</u> is used as a probe and re-retrieved from memory, however, the new composite vector contains context information from the term <u>capital</u>. The story is much the same for <u>Toronto</u>; the composite vector that is formed by using the composite vector for Toronto as a probe also contains <u>capital</u>. In sum, even though <u>Toronto</u> and <u>Brisbane</u> never occur in the same context, the model deduces that the terms are related because their contexts/documents have other words in common.

In the simulation to follow, the same idea was applied to a much larger corpus of text. The model encoded one year's worth of articles from an Australian newspaper. The corpus contains approximately 2 million words of text over about 30,000 articles.

Before I go on to discuss the simulation results, I will go over some details pertaining to the pre-processing on the co-occurrence matrix that occurs before retrieval. The first stage of pre-processing involved excluding terms on the basis of three properties.

Promiscuity: A word that occurs in almost every document carries little information about the message's topic (e.g., function words like the)

Monogamy: A word that occurs often but only in one document carries little information about what it could mean. In order to get a good representation of a word's meaning, there needs to be variety in the contexts/documents in which it appears. In the simulation reported below, a word needed to appear in at least two contexts to be encoded.

Celibacy: A word that virtually never occurs in the corpus of text does not carry much information about what it could mean. Only words that occurred at least twice in the corpus were included.

After filtering, the resultant matrix contained 86,125 unique terms taken from 38,525 newspaper articles. Following the example set by Landauer and Dumais (1997), the next step in pre-processing, the cells of the term-by-document matrix are transformed. First, each cell's frequency is transformed to its log. Then, the value is divided by a value that is a function of the entropy of the word across the contexts over which it appears. In weighting an entry by its entropy, each cell provides information about how uniquely a term is anchored to a context. More formally, each cell of a word's trace is transformed thusly:

$$W_i = \frac{\ln(f(W_i + 1)}{(-\sum p \ln(p))^{\beta}}$$

Where W is the raw frequency of word i in a context. The p is equal to the transformed frequency (i.e., the numerator of the term) of a word divided by sum of the frequencies of a word across contexts (C); i.e.,

$$p = \frac{\ln(f(W_i + 1))}{\sum_{c=1}^{C} \ln(f(W_i + 1))}$$

and β is an exponent that adjusts how strongly terms are anchored to the contexts. For the simulation reported below, the parameter was set to 2.

A simulation

To see whether the retrieval model could deduce semantic relationships among terms, the semantic representations for words from four categories were analysed using multidimensional scaling to determine if the model derived meaningful semantic representations.

Method

Thirty-two items representing four categories for words were used (places, domesticated animals, money, and modes of transport). The model retrieved the meaning vector for each word. Then, using the vector cosine as a measure of similarity between the meaning vectors of every possible pairing of words, a matrix of the similarities among them was formed.

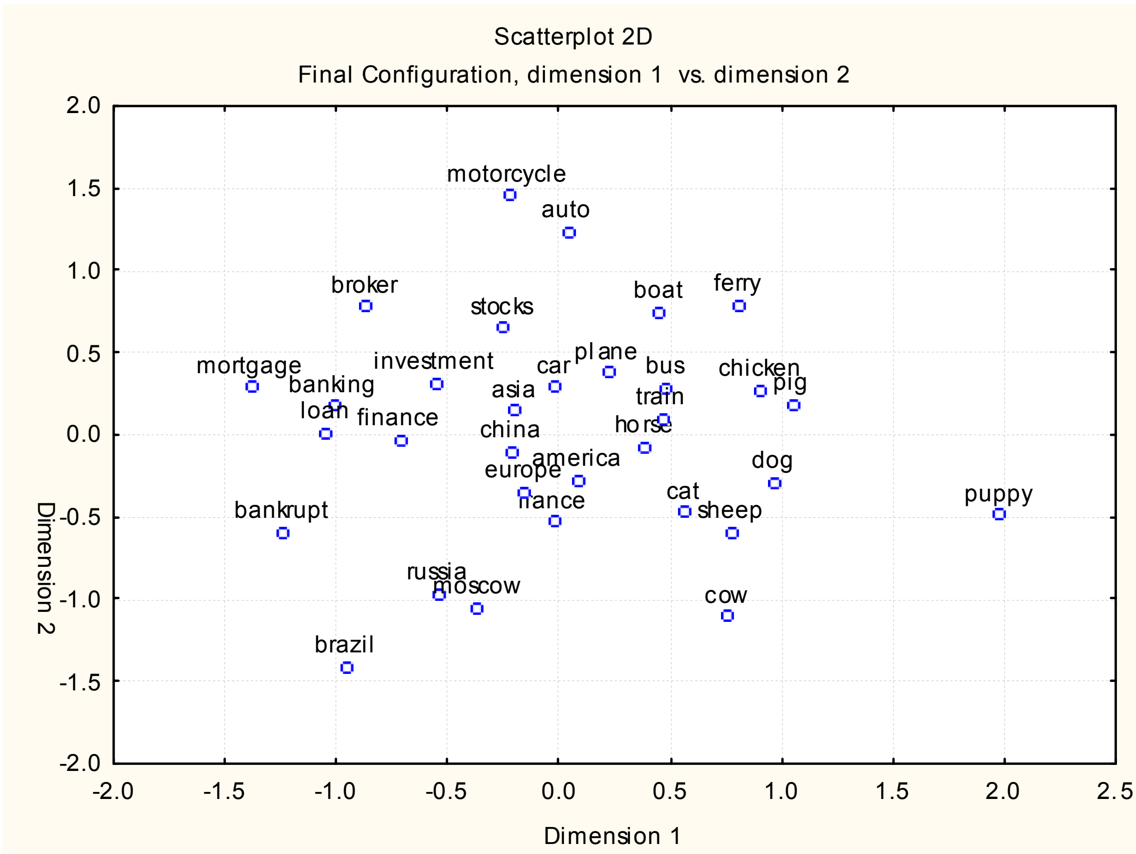


Figure 3. MDS solution for the full semantics model.

Results

The similarity matrix was analysed using multidimensional scaling (MDS). MDS is a technique that reduces coordinates from high to low dimensional space while simultaneously attempting its best to maintain the appropriate distance among points. For the simulation below, the MDS reduced the similarity matrix to two dimensions so that the terms could be plotted in (x, y) coordinates and easily visualised. Figure 3 shows the solution found by the reduction. As is clear in the figure, terms that are related to each other are, in general, clustered close together relative to unrelated concepts. Terms that are unrelated tend to be separated in semantic space. In another analysis of the output, I calculated the average similarity among terms within a category (excluding its similarity to itself) and between categories. Figure 4 shows clearly that, on average, terms are reliably more similar to other items within the same semantic category than with any other.

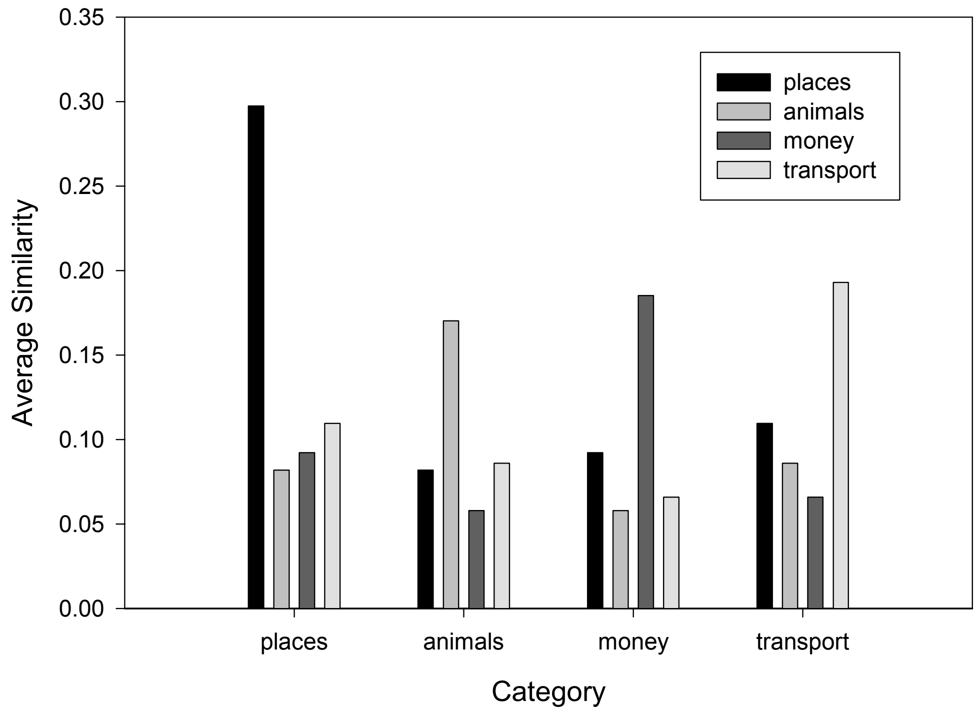


Figure 4. Mean within- and between-category similarities for each group of items.

Discussion

The model described above works for similar reasons as LSA. Local co-occurrences of words are not adequate to capture semantic relationships between words. In other words, while two words may appear in the same context, they may or may not be semantically related. By the same token, two words that never appear in the same document are not necessarily unrelated. In order to capture semantic relationships, higher-order relationships between words must be exploited. In the example from Figure 2, Toronto and Brisbane can be related concepts despite them not occurring in the same document. Their relationship develops because the documents they appear in share other words like capital, residents, population, and city. Put simply, the semantics model uses what it knows about a word's context, and makes a guess about what other contexts it might appear in, and the frequency with which it might appear in them. Albeit by different means, LSA does the same thing—it guesses how often a term occurs in each of several documents after the dimensionality of the original term-by-document matrix has been reduced by the SVD.

Deconstructing the model's output

Why do semantic relationships between words emerge from the model? Is the retrieval component responsible for them or is first-order co-occurrence information driving the similarities between terms? To answer this question, the same simulation was re-run with two versions of the model. In the first version, the retrieval stage was skipped and the similarity between pairs of terms was measured from the first-order, or raw, co-occurrence information. In other words, how often the words appeared in the same contexts. The second version of the model created a meaning vector entirely from second-order co-occurrence information. Specifically, the meaning vector (i.e., the composite vector from memory) created for a word in a pair excluded a copy of itself from memory and the other member of the pair.

After the two models had been run, the same MDS analysis was performed on resultant similarity matrices. Figures 5 and 6 show the MDS solutions for the first-order and second-order co-occurrence models, respectively. As is clear in the figures, both versions of the model seem to be clustering semantically related terms together. To show that both versions of the model seem to be clustering semantically related items, the average within- and between-group similarities were calculated for each. As can be seen in Figure 7, related terms are more likely to occur in the same context than different ones. Not a surprising finding—words that are related are often discussed in the same context. The important question, however, is whether the first-order co-occurrence information between the pairs of words used in the simulation is <u>responsible</u> for the model's ability to deduce that two words are semantically related. The issue is potentially fatal to the model; if true, the model would be unable to tell that two words were related unless they occurred in the same document. What is more, the model would assume that two words are related simply <u>because</u> they appear in some of the same documents.

Figure 8 contains the average within- and between-group similarities of words based entirely on second-order co-occurrence information. Note the similarity between the second-order model and full model. The two graphs are almost identical. That the graphs (and the MDS solutions) are essentially the same, suggests that the first-order co-occurrence information plays little if any role in creating the meaning. Instead, the meaning vector that is retrieved from the full model is almost entirely made up of second-order co-occurrence information. To illustrate the point, I plotted each item's average similarity to the items of each of the four groups for the full model against the first- and second-order models. The plot is shown in Figure 9. As can be seen in the figure, there is a perfect correspondence between the

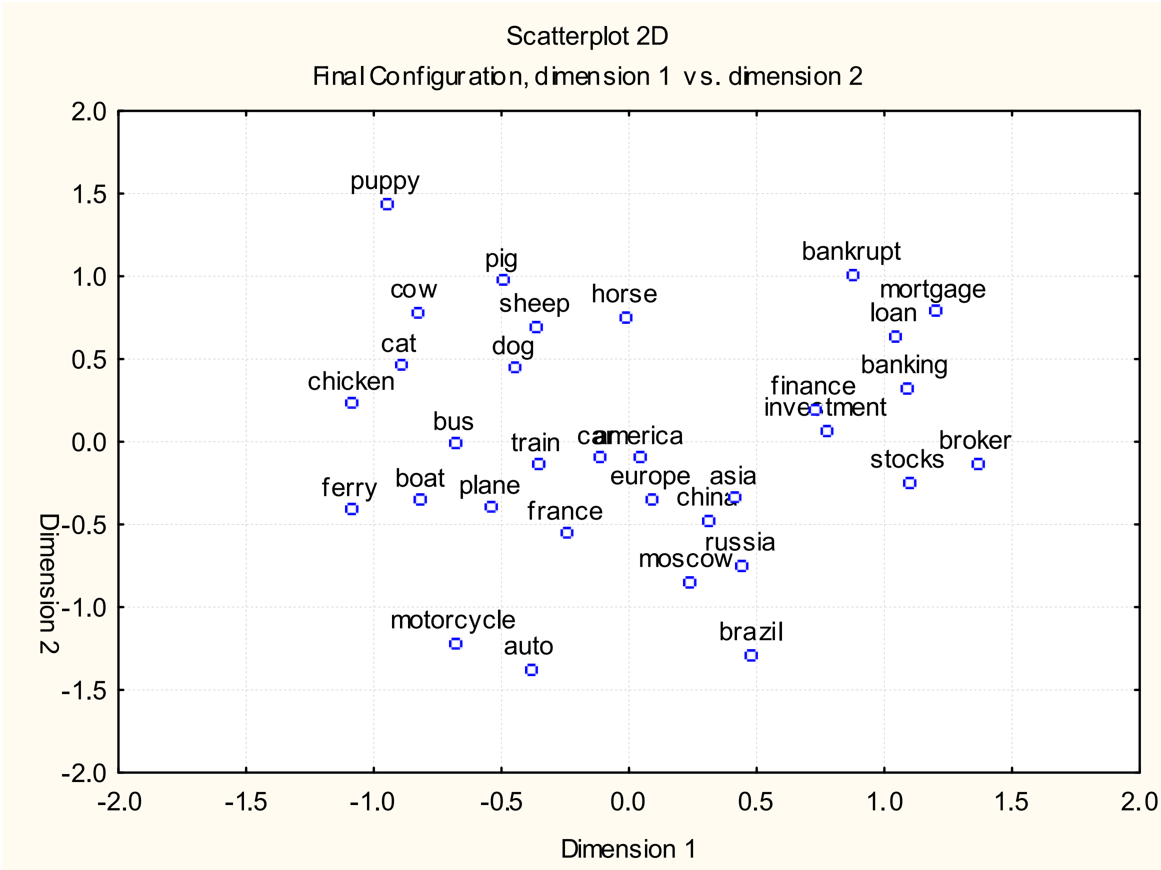


Figure 5. MDS solution for the first-order co-occurrence model.

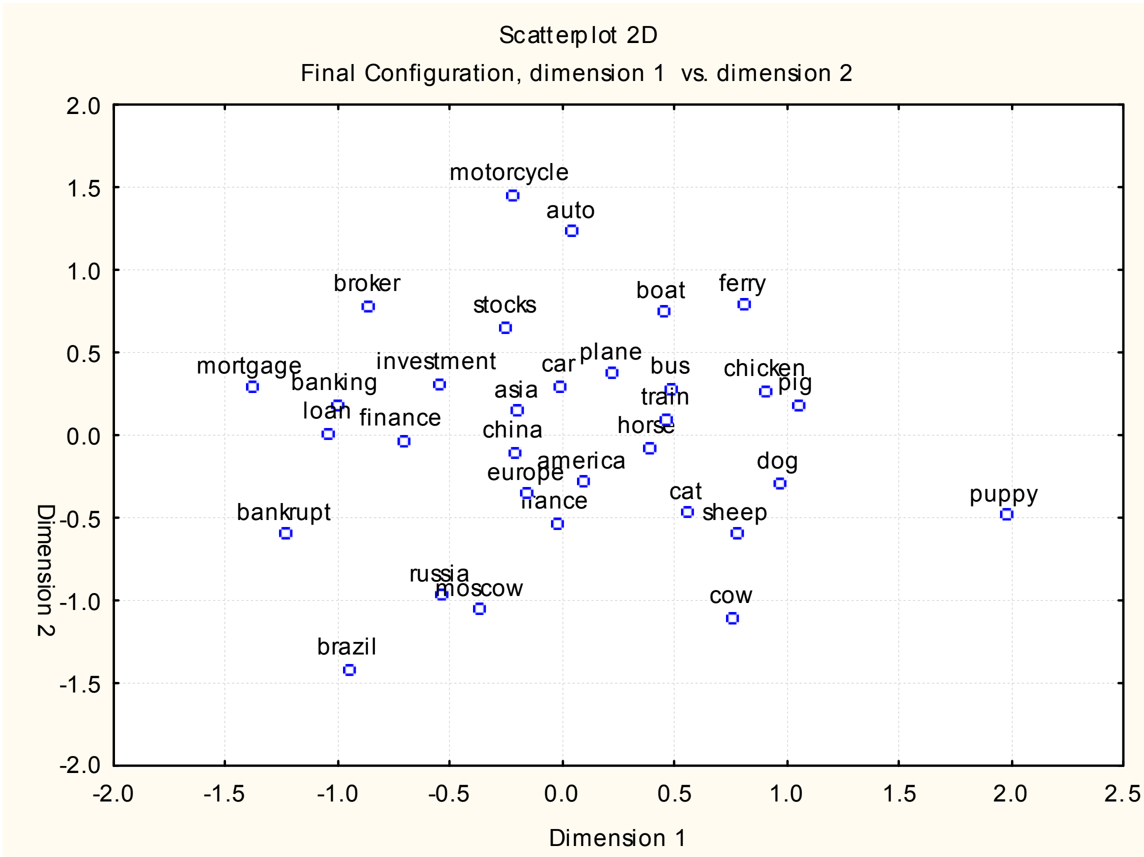


Figure 6. MDS solution for the second-order co-occurrence model

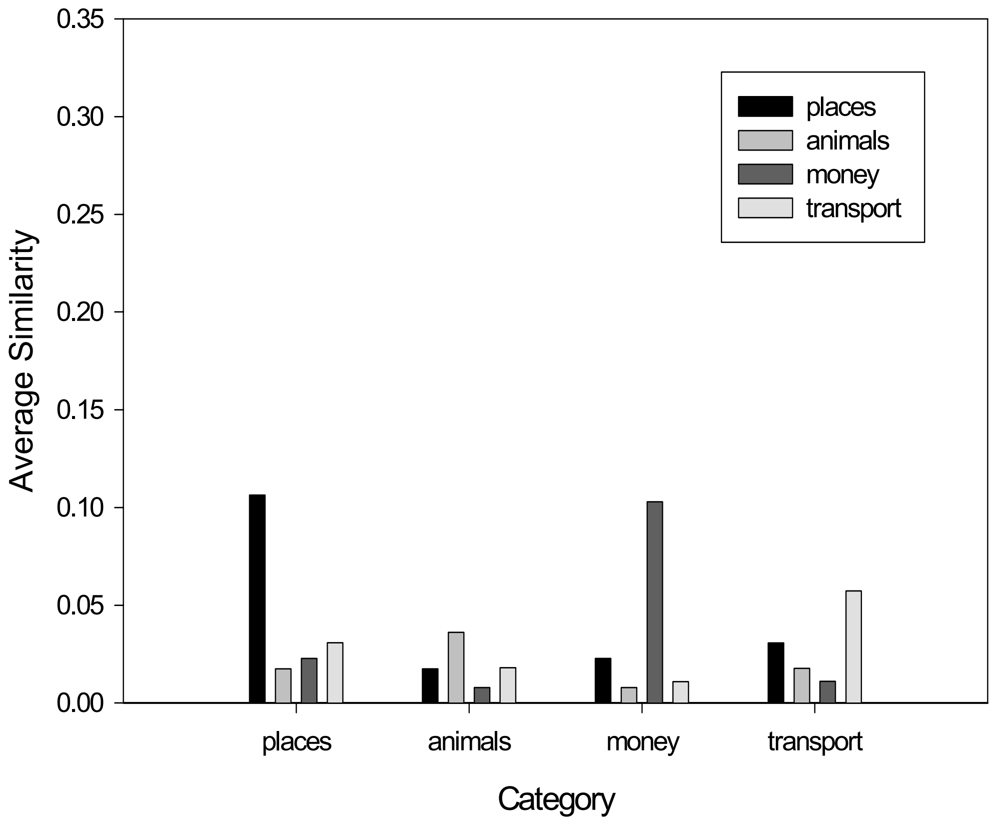


Figure 7 Average within- and between-group similarities of words based entirely on first-order co-occurrence information.

similarities of the full and second-order models. The first-order co-occurrence information does not contribute as consistently to the full model's meaning vector as does the higher-order information. The influence that the first-order co-occurrences information has on the meaning vector is overpowered by second-order information because, while the former represents the similarity of two context vectors in a pair, the latter's vectors includes the influence of thousands of memory traces that are summed during retrieval.

A final point to address in this section is the issue of whether the model <u>needs</u> to perform a retrieval operation at all. As the MDS solution in Figure 5 and the bar chart in Figure 7 show, the raw, or first-order, co-occurrence information about the terms used in the simulation seemed to be adequate for separating the concepts into semantic neighbourhoods. Why should we bother with the computationally expensive retrieval stage? While it is true that, in this case, related terms showed a greater tendency to co-occur in documents than unrelated terms, in the end, such raw co-occurrence information is not guaranteed to capture the similarity between terms. As mentioned at the beginning of the discussion section, many related terms will never occur in the same context. Hence, if the system relied solely on raw co-occurrence information, several relationships between terms would be undetected.

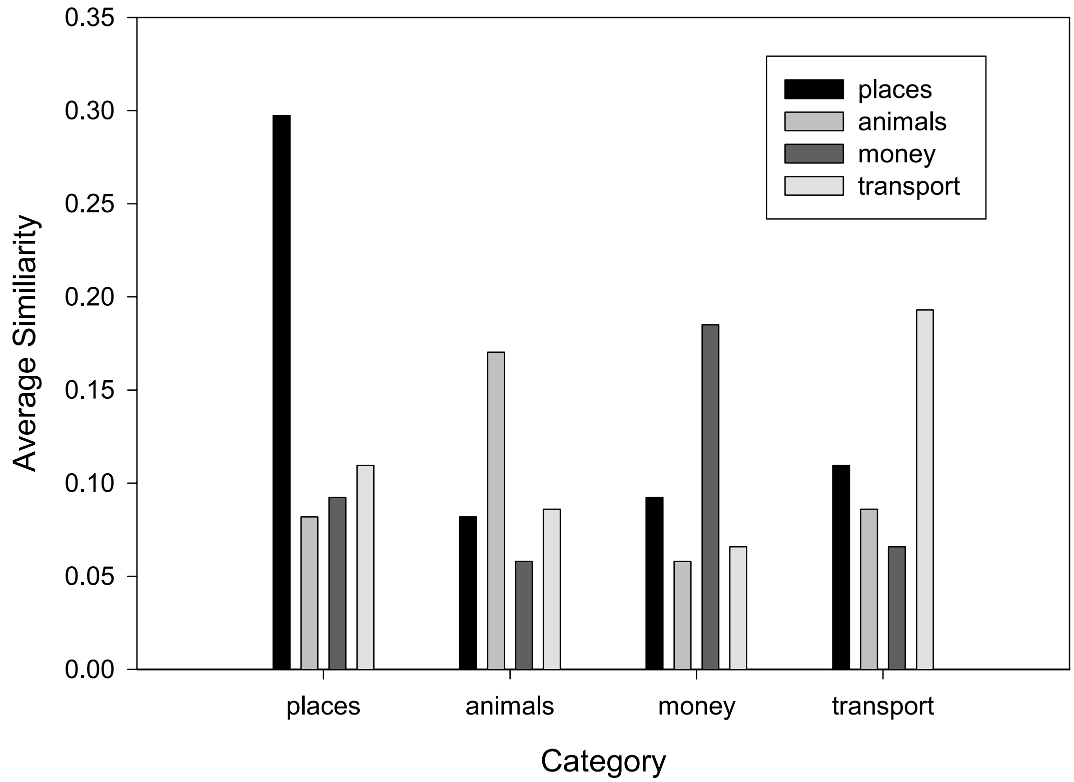


Figure 8 Average within- and between-group similarities of words based entirely on second-order co-occurrence information.

Implications of the ideas embodied in the semantics model

The semantics model represents a potential unification of formal models of episodic and semantic memory. The model was built to simulate how people form representations for the meanings of words they know—so-called <u>semantic memory</u>. As discussed above, however, with a couple of minor exceptions, the semantics model is, architecturally, almost identical to Minerva2 (Hintzman, 1984; 1986; 1988), a well-known model of episodic memory designed to simulate human performance in laboratory-based memory experiments. The similarity between the two models suggests that perhaps the same basic memory system underlies both forms of knowledge.

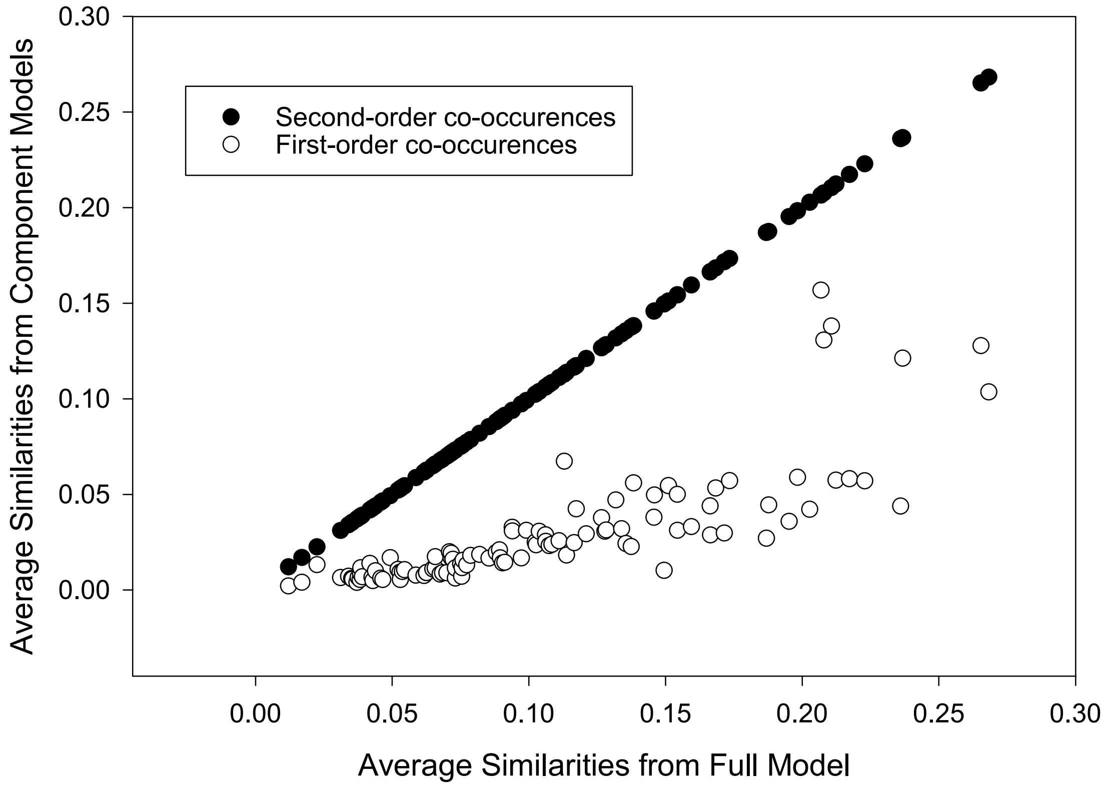


Figure 9. Each item's average similarity to the items of each of the four groups for the full model plotted against the same items in the first- and second-order co-occurrence models. .

From a psychological perspective, the model takes a unique perspective on the representation of semantic knowledge. In essence, it postulates that we <u>don't</u> represent semantic information. Instead, all that is required of memory is that it stores the contexts that are associated with the terms we encounter, and that a representation of the meaning of an item is <u>constructed</u> from the contextual information during retrieval. The idea that we store the contexts associated with an item is not controversial. Indeed, Dennis and Humphreys (2002) proposed a model of episodic memory that uses the interaction between experimental context and pre-existing contextual information in memory to explain several well-researched phenomena found in episodic memory tasks.

The idea that semantics are constructed rather than stored may also serve as an explanation for how a person's own definition of a word can change over time. Suppose a banker switches occupations to that of ferryboat captain. What does the word 'bank' mean to that person? I suspect that before taking a job on the ferry, "bank" was associated with money, but now is more associated with the part of the river his boat must avoid hitting. What changed? I believe that the new use of the word changed its contextual representation in memory; a change that, in turn, transformed the representation of its meaning. An appealing feature of my interpretation for the dynamic nature of meaning is that it requires no mechanism for changing it other than the addition of new contextual information to memory.

Conclusion

The semantics model was designed to retrieve the meanings of words from a matrix containing the frequency with which words occur across approximately 30,000 documents. It stands as a psychological theory of how people develop semantic representations for words, but it has possible applications in other more practical areas. The same operations on the same matrix could be used to retrieve the meaning of a document. While not that interesting from a psychological perspective, the system might have uses as a filtering system for machine-readable documents. For example, the system could be set up to cluster e-mails according to their topics. If an agency was interested in monitoring emails on a particular topic, they system would be able to single out those emails that were suspect. Furthermore, because emails are tagged with date, author and recipient information, the system could be used as part of a system that can uncover the social networks whose correspondence deserves attention. An attractive feature of the system is that it does not require any a priori knowledge of a language. The same system can develop semantic representations for any language—indeed, the system is so blind to the language it encodes it could develop semantic representations for whale and dolphin song if the materials could be parsed.

Whatever areas it comes to be used in as a tool, the semantics model described above represents a unique treatment of the problem of semantics as a field of psychological enquiry. It represents a first attempt at the unification of episodic and semantic memory models. In particular, it shows that the same basic architecture can be used to simulate behaviour in two fields of memory research that have almost always been studied separately. This final point is important because memory researchers have known for a long time that semantic information in memory can exert a marked influence on performance in tasks that examine episodic memory. The semantics model offers a framework to explain how the influence occurs.

References

1. Landauer, T. K. & Dumais, S. T. (1997). A solution to Plato's problem: The latent semantic analysis theory of the acquisition, induction, and representation of knowledge. Psychological Review, 104, 211-240.

2. Burgess C., Livesay, K., & Lund, K. (1998). Explorations in context space: Words, sentences, discourse. Discourse Processes, 25, 211-257.

3. Hintzman, D. L. (1984). MINERVA 2: A simulation model of human memory. Behaviour Research Methods, Instruments and Computers, 16, 96-101.\

4. Hintzman, D. L. (1986). "Schema abstraction" in a multiple-trace model. Psychological Review, 93, 411-428.

5. Hintzman, D. L. (1988). Judgments of frequency and recognition in a multiple-trace memory model. Psychological Review, 95*,* 528-551.

6. Dennis, S. & Humphreys, M.S. (2001). A context noise model of episodic word recognition, Psychological Review,. 108, 452–478.

7. Dumais, S.T. (1994). Latent Semantic Indexing (LSI) and TREC-2. In D. Harnan (Ed.) The second text retrieval conference (TREC2) (National Institute of Standards and Technology Special Publication 500-215, pp.105-116.

8. Dumais, S. T. & Nielson, J. (1992). Automating the assignment of submitted manuscripts to reviewers. In N. Belkin, P. Ingwesen, & A.M. Pejtersen (Eds.) Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. New York: Association for Computing Machinery.

9. Landauer, T.K., Foltz, P.W., & Laham, D. (1998). Introduction to Latent Semantic Analysis. Discourse Processes, 25, 259-284.

List of symbols/abbreviations/acronyms/initialisms

DND	Department of National Defence
LSA	Latent Semantic Analysis
MDS	Multidimensional Scaling

DOCUMENT CONTROL DATA SHEET

1a. PERFORMING AGENCY DRDC Toronto	2. SECURITY CLASSIFICATION UNCLASSIFIED Unlimited distribution -
1b. PUBLISHING AGENCY DRDC Toronto	

3. TITLE (U) Context as the building blocks of meaning: A retrieval model for the semantic representation of words
4. AUTHORS Peter J. Kwantes

5. DATE OF PUBLICATION April 1 , 2003	6. NO. OF PAGES 27

7. DESCRIPTIVE NOTES
8. SPONSORING/MONITORING/CONTRACTING/TASKING AGENCY Sponsoring Agency: Monitoring Agency: Contracting Agency : Tasking Agency:

9. ORIGINATORS DOCUMENT NO. Technical Report TR 2003-061	10. CONTRACT GRANT AND/OR PROJECT NO.	11. OTHER DOCUMENT NOS.

12. DOCUMENT RELEASABILITY Unlimited distribution
13. DOCUMENT ANNOUNCEMENT Unlimited announcement

14. ABSTRACT

(U) A computational model of semantic memory is described. Based on simple principles borrowed from a computational account of episodic memory, it is shown that a memory model that is exposed to a large corpus of language can develop representation for words that look like their ‘meanings’.

(U) Ce rapport décrit un modèle informatique de mémoire sémantique. Partant de simple principes empruntés à la description computationnelle d’une mémoire épisodique, on démontre qu’un modèle de mémoire exposé à un vaste corpus de mots peut former une représentation de mots qui ressemble au sens de ces mots.

15. KEYWORDS, DESCRIPTORS or IDENTIFIERS

(U) model; semantics; computational; memory; meaning